

# Advanced Java Programming

## Week 3 Topic Outline

### Abstract Classes and Interfaces

1. Abstract vs. concrete classes
2. Abstract methods and classes
3. Interfaces and can-do relationships
  - a. The interface contract
  - b. Programming to an interface, not an implementation
4. Favor composition over inheritance

### Javadoc Documentation

1. Javadoc lets us view the "interface" for a class
2. Preconditions and post-conditions
3. Javadoc comments
4. `@author`, `@version`, `@param`, `@return`, `@throws`
5. Generating documentation with the `javadoc` tool
6. Google "javadoc tutorial" for more information

### Packages

1. Java's package system
  - a. Correspondance to file system
  - b. Non-hierarchical package names
2. Importing classes
  - a. Importing by name
  - b. Wildcard importing
  - c. The `java.lang` package
3. Referring to objects by package name
4. Importing static members

### The Java API

1. Navigating the API
2. The `StringBuilder` class
3. The `ArrayList` class

## Wrapper Classes

1. Wrapper class  $\Rightarrow$  class that wraps data to extend functionality
2. Store primitive data in `ArrayList` with wrapper classes
  - Can't create `ArrayList` of primitive type because raw `ArrayList` is a list of `Object`s, and the supplied types are translated by the compiler as downcasts from `Object`.
3. Autoboxing
  - Occurs when primitive is passed to method expecting wrapper
  - Occurs when primitive is assigned to a variable of the wrapper type
4. Unboxing
  - Occurs when wrapper is passed to method expecting primitive
  - Occurs when wrapper is assigned to a variable of the primitive type